Analysis of Machine Learning Based Malicious URL Detection Methods

Project By: Sankalp Talankar, Rajan Patel and Rahul Porwal

Guided By: Dr. Patrick Traynor

Abstract

Machine learning approaches have received more attention in recent years to improve the generality of malicious URL detectors. Most of the papers for these models use different datasets and metrics to evaluate their performance, making it difficult to compare the models. The purpose of this paper is to use a standardized dataset and metric to compare some of the latest machine learning algorithms for malicious URL detection. We are using the URL dataset provided by the University of New Brunswick consisting of around 610,000 URLs as a standard dataset to evaluate our models. Since it is much more damaging if a malicious URL is not detected as compared to a benign URL that is classified as malicious, it is more relevant to analyze how well the model can detect the malicious URLs. We decided to use AUPRC (Area under Precision- Recall curve) as the standard evaluation metric. Our analysis shows that deep learning models like URLNet perform better than traditional machine learning models because machine learning models use manually engineered features and they do not generalize well on data where such features are not present. So, while machine learning models are currently more popular, moving forward, deep learning models will be more prevalent.

Elements of a URL

https://www.cise.ufl.edu/class/cnt5410fa22/schedule.html

Elements of URL	Description
https	Protocol used to access the web server resources
cise.ufl.edu	Domain name used to find resource on the internet and is divided into 3 levels: subdomain (cise), secondary level domain(ufl), and top level domain(edu).
Cnt5410fa22	Resource Directory
schedule.html	Requested web page

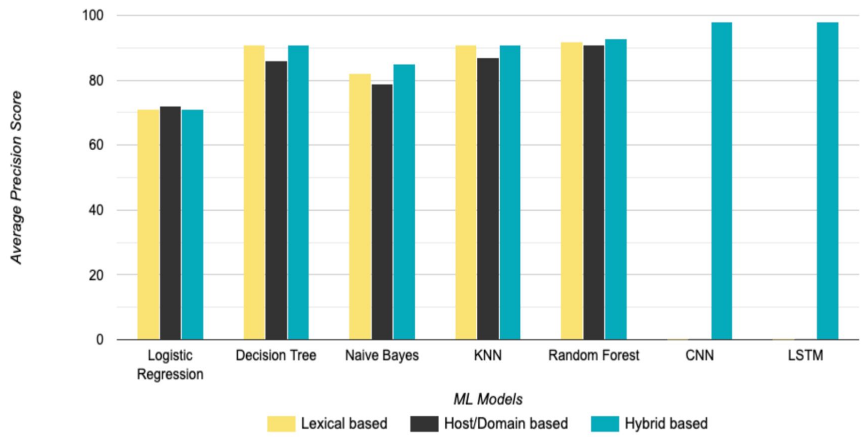
Detection Techniques

1. Blacklisting
2. Rules-Based Systems
3. Machine Learning
4. Deep Learning

Features Used by the Models

Type of Features	Pros	Cons
Lexical Features	Provides simple, efficient clues about a URL's potential harm.	Can be easily manipulated by attackers, leading to a high rate of false negatives.
Host or Web Server based Features	Provides detailed information about a URL's origins, including the domain name, IP address, and server location.	Require multiple network requests to the server hosting the URL, which can be resource-intensive.
Content based features	Comprehensive way to identify potentially harmful URLs. Features, such as text, images, on a page, provide a wealth of information about a URL's nature and intent.	Resource-intensive and time-consuming. Gathering these features involves downloading and analyzing the entire contents of a page.
Reputation based features	Identifies previously flagged malicious URLs, effectively detecting known threats and protecting users.	May not be effective against new or unknown threats.

Results



Key Findings

In the field of malicious URL detection, lexical features are commonly used as input features for machine learning models. However, these features can be easily manipulated by attackers, making it challenging for the model to accurately identify malicious URLs. To address this limitation, it is recommended to use hybrid features as input features for the model. Additionally, adversaries are becoming more adept at concealing the maliciousness of their URLs through techniques such as obfuscation and encryption. Manually engineered features are not able to effectively handle these types of URLs, therefore new techniques such as deep learning are required to improve the accuracy of malicious URL detection.

Conclusion

Our experiments have confirmed our initial hypothesis that deep learning models would outperform traditional machine learning techniques in the task of malicious URL detection. Specifically, convolutional neural networks (CNNs) and long short-term memory (LSTM) networks were found to perform better than other traditional machine learning approaches. Furthermore, we observed that using a combination of URL features (known as hybrid features) improved performance compared to using lexical features alone, highlighting the importance of careful feature engineering in this domain. Deep learning models are well-suited to this task because they do not require manual feature engineering and can generalize well to new data